

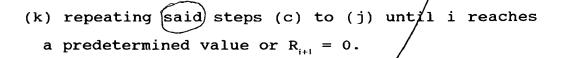
16

15

1. In a system for digital information process, a method for finding a quotient $Q = a_0 a_1 a_2 \dots a_b$ from a divisor $Y = y_1 y_2 \dots y_n$ and a dividend $X = x_1 x_2 \dots x_a$, comprising the following steps of:

- (a) aligning the first non-zero bit of X with the first non-zero digit of Y/;
- (b) defining a signed-digit partial remainder series R_i where $R_0 = Y$, a first sign series of the partial remainder S_i where $S_0 \neq 0$, a second sign series of the partial remainder S_{r_i} , and a counter i beginning from zero.
 - (c) subtracting X from R_i which yields next signed-digit partial remainder R_{i+1} ;
 - (d) setting the sign of R_{i+1} to $S_{r_{i+1}}$;
 - (e) setting the result of exclusive-OR of S_i and S_{n+1} to the true sign of the next remainder S_{i+1} ;
- 20 (f) setting $a_i \neq 0$ 1 if $S_{i+1} = 0$ or $R_{i+1} = 0$;
 - (g) setting $a_i / to 0$ if $S_{i+1} = 1$;
 - (h) inverting the signs of all digits of R_{i+1} if $S_{i+1} = 1$;
 - (i) shift R left by one bit;
- 25 (j) adding $\int 1$ to i; and





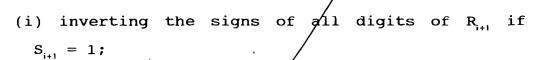
- 2. In a system for digital information process, a method for finding a signed magnitude quotient $Q_2 = a_s a_0 \cdot a_1 a_2 \cdot \dots \cdot a_b$ from a signed divisor $Y_s = y_s \cdot y_1 y_1 \cdot \dots \cdot y_n$, and a signed dividend $X_s = x_s \cdot x_1 x_2 \cdot \dots \cdot x_s$, comprising the following steps of:
 - (a) obtaining a from the result of exclusive-OR of y and x;
 - (b) defining a divisor $Y = y_1 y_2 \dots y_n$, a dividend $X = x_1 x_2 \dots x_n$, a signed-digit partial remainder series R_i where $R_0 = Y$ a first sign series of the partial remainder S_i where $S_0 = 0$, a second sign series of the partial remainder S_{r_i} , and a counter i beginning from zero;
 - (c) aligning the first non-zero bit of X with the first non-zero digit of Y;
 - (d) subtracting x from R which yields next signeddigit partial remainder R;
 - (e) setting the sign of R_{i+1} to $S_{r_{i+1}}$;
 - (f) setting the result of exclusive-OR of S_i and S_{n+1} to the true sign of the next remainder S_{i+1} ;
 - (g) setting A_{i} to 1 if $S_{i+1} = 0$ or $R_{i+1} = 0$;
 - (h) setting/a to 0 if $S_{i+1} = 1$;

10

15

20

25



- (j) shift R_{i+1} left by one bit;
- (k) adding 1 to i; and
- 5 (1) repeating said steps (d) to (k) until i reaches a predetermined value or $R_{i+1} = 0$.